REMARKS

In the Office Action[1], the Examiner rejected claims 23, 25, and 26 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,991,893 by Snider ("*Snider*") in view of U.S. Patent No. 5,675,790 by Walls ("*Walls*"); rejected claims 30 and 32-34 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,535,364 by Resman et al. ("*Resman*") in view of *Walls*; and objected to claims 24 and 31 as being dependent upon a rejected base claim, but allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Applicant has amended claim 23 to correct a grammatical error. Claims 17-34 are pending, and claims 17-22 and 27-29 have been withdrawn.

I. Regarding the rejection of claims 23, 25, and 26 under 35 U.S.C. § 103(a) as being unpatentable over *Snider* in view of *Walls*

Applicant respectfully requests that the Examiner reconsider and withdraw the rejection of claims 23, 25, and 26 because a *prima facie* case of obviousness has not been established with respect to these claims.

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In*

[1] The Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). M.P.E.P. § 2142, 8th Ed., Rev. 2 (May 2004), p. 2100-128.

A *prima facie* case of obviousness has not been established because, among other things, neither *Snider* nor *Walls*, taken alone or in combination, teach or suggest each and every element recited by Applicant's claims.

Claim 23 recites a system including, for example:

> an area of a memory designated for coordinating the assignment of the memory to one or more threads requiring access to the memory, wherein the area includes usage information reflecting usage of the memory; and
>
> a processor for performing a protocol for serializing access to the memory by the one or more threads based on the usage information, wherein the protocol allows a first thread to access a first designated block of the memory while another thread requests and secures access to another block of the memory.

(emphasis added). *Snider* does not teach or suggest at least these elements. *Snider* discloses a "virtually reliable shared memory (VRSM) software layer 101 [that] ... localizes the fail-safe protocols for having a particular block of shared memory 106 be accessible to different nodes 104 of the hardware layer 102" (col. 4, line 62 - col. 5, line 1). The Examiner states that "the allocation table which is part of the VRSM layer 101 is a memory location accessed by multiple threads" (Office Action at page 8). Applicant respectfully disagrees.

The Examiner states that nodes 104 correspond to the claimed threads (Office Action at page 3). However, nodes 104 access memory 106 (*Snider*, col. 4, lines 65-67). These nodes do not access the VRSM layer 101. The Examiner asserts that the "assignment of memory to one or more threads 104 requir[es] access to the memory 106" (emphasis added) (Office Action at page 3). Even assuming that the Examiner is

correct when stating that the VRSM layer 101 includes "usage information reflecting usage of the memory" (Office Action at page 3), VRSM layer 101 is not accessed by nodes 104. *Snider* depicts two distinct memories 101 and 106 (Fig. 1). However, the nodes 104 only access the memory 106 and there is no teaching that these nodes access the VRSM layer 101.

Snider explicitly states that this "layer [101] localizes fail-safe protocols for having a particular block of shared memory 106 be accessible to the different nodes 104" (col. 4, lines 65-67). Therefore, *Snider* does not teach "an area of a memory designated for coordinating the assignment of the memory to one or more threads requiring access to the memory, wherein the area includes usage information reflecting usage of the memory," as recited in claim 23.

The Examiner correctly notes that *Snider* does not teach "serialized operation by the processor/controller to allow a first thread to access a first designated block of the memory while another thread requests and secures access to another block of the memory" (Office Action at page 3). However, the Examiner relies on *Walls* to teach this limitation.

Walls does not cure the deficiencies of *Snider*. *Walls* discloses that "[i]n systems where there is more than one processor, it is possible for two users to make simultaneous requests to dynamic memory. It is necessary for an operating system to prevent the allocation of the same sections of dynamic memory to two or more processes" (col. 2, line 67 - col. 3, line 4). In *Walls*, by "allocating pool memory quickly, the consumer will more quickly release serialized access to the pool structures and so reduce the delay to other potential customers" (col. 4, lines 64-67). Therefore, if pool

memory is allocating by a first user, any access to that memory by a second user will result is an access delay until the first user releases access to the memory. *Walls* does not teach "an area of a memory designated for coordinating the assignment of the memory to one or more threads requiring access to the memory, wherein the area including usage information reflecting usage of the memory," as recited in claim 23.

Accordingly, *Snider* and *Walls* fail to establish a *prima facie* case of obviousness with respect to claim 23, at least because the references fail to teach each and every element of the claim. Claims 24-26 depend from claim 23 and are thus also allowable for at least the same reasons as claim 23.

II. Regarding the rejection of claims 30 and 32-34 under 35 U.S.C. § 103(a) as being unpatentable over *Resman* in view of *Walls*

Applicant respectfully requests that the Examiner reconsider and withdraw the rejection of claims 30 and 32-34 because a *prima facie* case of obviousness has not been established with respect to these claims.

Claim 30 recites a system including, for example:

> allocating to a first process, without accessing an operating system, a first block of a memory that has a size designated by a user; and
> allocating to a second process, without accessing an operating system, a second block of the memory that has a size designated by the user while the first process is accessing the first block of memory.

(emphasis added). In *Resman*, "I/O control module 16 receives data from a host processor, converts it to a form suitable for storage and transmits it, via bus 18, to a RAM 20" (col. 3, lines 6-8). Accordingly, *Resman* does not teach a user performing any designations. The only reference to a user in the context of memory is found in col. 1, lines 65-66, which states, "[a]s an example, a printer has available to it a set amount of

RAM depending upon the amount installed by the user." This statement merely shows a user may install a certain amount of RAM, but does not suggest "allocating to a first process, without accessing an operating system, a first block of a memory that has a size designated by a user," as recited in claim 30.

Resman discloses a memory allocation system that allows applications to request and receive RAM space from a free RAM pool. If no space is available, an application is allowed access to an AI/O RAM pool (col. 3, lines 29-49). I/O tasks are allocated space in an I/O fixed buffer pool. An I/O task may be allocated space in the AI/O RAM pool only when a certain amount of space is available in the free RAM pool (col. 3, lines 50-62). Thus, contrary to the Examiner's assertions, *Resman* does not teach or suggest "allocating to a first process, without accessing an operating system, a first block of a memory that has a size designated by a user."

Resman does not state that the size of any portion of the RAM pools is designated by a user. On the contrary, *Resman* states that requests for RAM allocation are caused by an application running in CPU 14 (col. 3, lines 29-30). This software-based allocation process does not provide for user input, much less input to designate a size of a block of memory to be allocated to a process. Further, the mere fact that *Resman* allows an I/O task to access memory (under certain conditions) does not bolster the Examiner's position. As explained, *Resman* does not mention or disclose a user-designated size of block of memory that is allocated without accessing an operating system to a first or second process. Therefore, *Resman* does not teach or suggest "allocating to a first process, without accessing an operating system, a first block of a memory that has a size designated by a user" and "allocating to a second

process, without accessing an operating system, a second block of the memory that has a size designated by the user while the first process is accessing the first block of memory," as recited in claim 30.

The Examiner correctly notes that *Resman* does not teach "serialized operation by allowing a first thread to access a first designated block of the memory while another thread requests and secures access to another block of the memory" (Office Action at page 6). However, the Examiner relies on *Walls* to teach this limitation.

Walls does not cure the deficiencies of *Resman*. As previously stated, *Walls* discloses that by "allocating pool memory quickly, the consumer will more quickly release serialized access to the pool structures and so reduce the delay to other potential customers" (col. 4, lines 64-67). Therefore, if pool memory is allocated to a first user, any access to that memory by a second user will result is an access delay until the first user releases access to the memory. *Walls* thus does not teach "allocating to a first process, without accessing an operating system, a first block of a memory that has a size designated by a user" and "allocating to a second process, without accessing an operating system, a second block of the memory that has a size designated by the user while the first process is accessing the first block of memory," as recited in claim 30.

Accordingly, *Resman* and *Walls* fail to establish a *prima facie* case of obviousness with respect to claim 30, at least because the references fail to teach each and every element of the claim. Claims 31-34 depend from claim 30 and are thus also allowable for at least the same reasons as claim 30. The rejections under 35 U.S.C. § 103(a) are improper and should be withdrawn.

III. Conclusion

In view of the foregoing remarks, Applicant respectfully requests reconsideration and reexamination of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

Jeffrey A. Berkowitz

Dated: May 26, 2006

By: Robert E. Cervenka #27,432
Jeffrey A. Berkowitz
Reg. No. 36,743